AsymDex: Leveraging Asymmetry and Relative Motion in Learning Bimanual Dexterity

Zhaodong Yang[1], **Yunhai Han**[1], **Harish Ravichandar**[1]
[1]Georgia Institute of Technology
halyang, yhan389, harish.ravichandar@gatech.edu

Abstract: We present *Asymmetric Dexterity (AsymDex)*, a novel reinforcement learning (RL) framework that can efficiently learn asymmetric bimanual skills for multi-fingered hands without relying on demonstrations, which can be cumbersome to collect. Two crucial ingredients enable AsymDex to reduce the observation and action space dimensions and improve sample efficiency. First, AsymDex leverages the natural asymmetry found in human bimanual manipulation and assigns specific and interdependent roles to each hand: a *facilitating hand* that moves and reorients the object, and a *dominant hand* that performs complex manipulations on said object. Second, AsymDex defines and operates over *relative* observation and action spaces, facilitating responsive coordination between the two hands. Further, AsymDex can be easily integrated with recent advances in grasp learning to handle both the object acquisition phase and the interaction phase of bimanual dexterity. Unlike existing RL-based methods for bimanual dexterity, which are tailored to a specific task, AsymDex can be used to learn a wide variety of bimanual tasks that exhibit asymmetry. Detailed experiments on four simulated asymmetric bimanual dexterous manipulation tasks reveal that AsymDex consistently outperforms strong baselines that challenge its design choices, in terms of success rate and sample efficiency. The project website is at https://sites.google.com/view/asymdex-2024/.

Keywords: Dexterous Manipulation, Bimanual Manipulation

1 Introduction

We tackle the challenge of learning bimanual dexterous manipulation skills on multi-fingered hands using reinforcement learning (RL). Bimanual skills are crucial for robots operating in human environments as they allow for more complex and flexible manipulation compared to a single hand [1–7]. While learning dexterous manipulation skills on a single hand presents numerous challenges [8–14], learning bimanual dexterity can be significantly more challenging due to the higher-dimensional state and action spaces and the need to coordinate and synchronize the movement of two hands [1].

To circumvent the challenges introduced by the high-dimensional state and action spaces of bimanual dexterous manipulation, we take inspiration from how humans approach this challenge. Humans exhibit a natural *asymmetry* in how we use each of our hands when we perform most bimanual tasks. Specifically, we tend to use one hand to reposition and reorient an object being manipulated so as to make it easier for the other hand to achieve the desired manipulation objectives. While the asymmetric assumption might seem restrictive at first glance, rich bodies of work in human biomechanics and evolution reveal its significance and necessity [15–18]. Evolutionary biologists posit that such handedness evolved in humans and great apes to meet the escalating cognitive demands of tool use and complex manipulation [19]. Indeed, a large class of real-world bimanual tasks admit this asymmetry (e.g., attachment, detachment, assembly, pouring).

A key insight that we leverage is that the natural asymmetry between the two hands can reduce the dimensionality of the bimanual dexterous manipulation, and in turn improve effectiveness and sample efficiency. Using this insight, we contribute a novel learning framework, dubbed *Asymmetric Dexterity (AsymDex)*, that can efficiently learn asymmetric bimanual dexterous manipulation tasks based on RL (see Fig. 1 for a block diagram). Note that AysmDex does not require the cumbersome collection of demonstrations and can learn bimanual skills only using reinforcement learning.

8th Conference on Robot Learning (CoRL 2024), Munich, Germany.

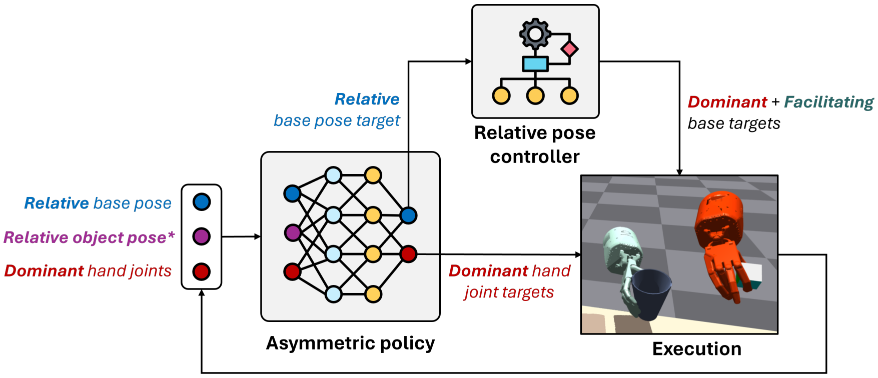


Figure 1: Our approach (AsymDex) efficiently learns asymmetric bimanual dexterous manipulation skills based on reinforcement learning by effectively leveraging i) the natural asymmetry in the hands' roles and ii) relative state and action spaces that encourage synchronization.

AsymDex has two crucial ingredients. First, it introduces asymmetry by defining a *dominant hand* and a *facilitating hand*. While the facilitating hand learns to reposition and reorient the object, the dominant hand learns complex manipulation skills (including in-hand manipulation). Note that there's no relative movement between the grasped object and the facilitating hand. As a result, AsymDex holds the fingers of the facilitating hands in the grasping pose and only controls the 6D motion of its base. On the other hand, AsymDex learns to control both the base and fingers of the dominant hand. Second, AsymDex reasons about and controls the *relative motion* between the dominant and facilitating hands. We define relative observation and action spaces that incentivize responsive coordination between the two hands without resorting to explicit time-dependence.

We also leverage the observation that bimanual manipulation in practice is composed of two distinct phases: i) the *acquisition phase* in which objects are grasped from surfaces, and ii) the *interaction phase* where the two hands coordinate to perform the bimanual task. Unlike many existing methods that ignore the acquisition phase, we show that this decomposition enables AysmDex to be seamlessly integrated with learned grasping policies to enable fluent execution.

In summary, we contribute AsymDex – a novel framework for asymmetric bimanual dexterity that leverages the natural asymmetry in hand roles and relative observation and action spaces. We evaluate AsymDex on four complex bimanual dexterous tasks (adapted from BiDexHand [20]) and compare against strong baselines that challenge its design choices. Our results show that AsymDex consistently outperforms the baselines in terms of success rate and sample efficiency.

2 Related Work

In this section, we contextualize our contributions within relevant sub-fields.

Learning Bimanual Manipulation: Several existing methods focus on learning bimanual skills, but are often limited to simple end-effectors. Imitation learning (IL) based approaches have been particularly successful in bimanual manipulation [3, 21–23], and have led to novel and low-cost infrastructure to collect bimanual manipulation data [4, 24]. These approaches rely on demonstrations to provide the necessary supervision to learn effective coordination strategies. Reinforcement learning (RL) has also been shown to be successful in learning bimanual manipulation skills [25–28]. These methods implicitly incentivize coordination by learning to optimize reward functions that favor task success and efficiency. In contrast to all of these works that only consider parallel jaw grippers, AsymDex learns bimanual dexterous manipulation skills involving multi-fingered hands.

Asymmetry in Bimanual Manipulation: Motivated by the asymmetry in how humans use their two hands (referred to as *role-differentiated bimanual manipulation* [16–18, 29]), recent works assign different roles to each robot hand in the bimanual system [2, 30–33]. However, some of these approaches restrict the role of the facilitating hand to stabilizing the object while the dominant hand manipulates it [2, 30, 31]. In contrast, AsymDex allows the facilitating hand to reposition and reorient the object *simultaneously* as the dominant hand executes its role. Importantly, unlike AsymDex, all these prior methods are limited to parallel jaw grippers.

Learning Dexterous Manipulation: Learning dexterous manipulation skills involves addressing numerous challenges due to high dimensional state and action spaces and highly nonlinear dynamics. Recent works have tackled these challenges using imitation learning (IL) or reinforcement learning (RL) and demonstrate impressive performance [8–12, 14, 34–37]. However, IL-based methods, including those that combine RL and IL [8, 10], rely either on complex infrastructure and retargeting methods to collect demonstrations [8, 14, 35, 38, 39] or pre-trained expert policies [11, 12, 37]. On the other hand, RL-based methods do not share these constraints as they learn skills via reinforcement. However, RL methods tend to require single amounts of exploration even for unimanual dexterous manipulation [9, 10, 34, 36]. As we show in our experiments, naive application of RL-based methods is not effective for bimanual dexterous manipulation due to the increased dimensionality and the need for coordination. We also show that AsymDex is able to efficiently learn complex bimanual dexterous manipulation skills using RL without relying on demonstrations.

Learning Bimanual Dexterous Manipulation: A few recent studies have focused on learning bimanual dexterity. Some of these methods require the collection of expert demonstrations [6] and suffer from the same limitations we discussed earlier for IL-based methods that use parallel jaw grippers. To circumvent the need for collecting demonstrations, recent efforts have led to methods that only leverage RL and yet are capable of learning impressive bimanual manipulation skills, such as playing the piano [7], twisting lids off containers [40], and dynamic handover [5]. While these methods are specifically designed to solve a particular task, AsymDex is capable of efficiently learning different bimanual dexterous manipulation tasks.

3 Learning Asymmetric Bimanual Dexterous Manipulation Skills

In this section, we formulate the problem of asymmetric bimanual dexterous manipulation and introduce the different elements of our approach (AsymDex).

3.1 Preliminaries

Consider the problem of bimanual dexterous manipulation, in which two multi-fingered hands coordinate to manipulate up to two objects. Formally, this problem can be defined as a Partially-Observable Markov Decision Process (POMDP) $\mathcal{M} = (\mathcal{S}, \mathcal{Z}, \mathcal{A}, \mathcal{R}, \mathcal{P})$, where $\mathcal{S} \in \mathbb{R}^n$ is the state space, $\mathcal{Z} \in \mathbb{R}^m$ is the observation space, $\mathcal{A} \in \mathbb{R}^u$ is the action space, $\mathcal{R} : \mathbb{R}^m \times \mathbb{R}^u \to \mathbb{R}$ is the reward function, and $\mathcal{P} : \mathbb{R}^n \times \mathbb{R}^u \to \mathbb{R}^n$ is the environment dynamics. Note that we do not assume access to any demonstrations. Instead, we tackle of challenge of learning purely based on reinforcement. Given this formulation, the problem boils down to learning a policy $\pi : \mathcal{Z} \to \mathcal{A}$ that maximizes the expected discounted cumulative reward $E_\pi[\Sigma_{t=0}^{T-1} \gamma^t \mathcal{R}(z(t), a(t))]$.

Observation and Action Space: The observation space $\mathcal{Z}$ is composed of hand and object measurements. At step t, $z(t)$ contains $\xi_1^b(t) \in \mathbb{R}^6$, $\xi_2^b(t) \in \mathbb{R}^6$, $\xi_1^h(t) \in \mathbb{R}^{n_1}$, $\xi_2^h(t) \in \mathbb{R}^{n_2}$, $o_1(t) \in \mathbb{R}^6$, and $o_2(t) \in \mathbb{R}^6$, where ξ_1^b and ξ_2^b denote the first and the second hand bases' $6D$ poses, ξ_1^h and ξ_2^h denote the first and the second hands' joint states (e.g., the palm and fingers), and o_1 and o_2 represents the $6D$ poses of either two objects (e.g., stacking two cups), or two parts of the same object (e.g., bottle and bottle cap). The action at Step t is given by $a(t) = [\hat{\xi}_1^b(t), \hat{\xi}_2^b(t), \hat{\xi}_1^h(t), \hat{\xi}_2^h(t)]$, composed of the target base poses ($\hat{\xi}_1^b$ and $\hat{\xi}_2^b$) and target joint positions ($\hat{\xi}_1^h$ and $\hat{\xi}_2^h$) for both hands. These actions are fed to a PD tracking controller to actuate both hands.

Note that our primary contributions pertain to the *interaction phase* of bimanual dexterous manipulation, in which two dexterous multi-fingered hands coordinate to complete the task after having grasped the necessary object(s). Most existing works focus solely on the interaction phase [5, 40, 41]. In Section. 3.6, we discuss how our approach can be extended to also tackle the *acquisition phase*, in which the hands learn to grasp objects before coordinating.

3.2 A Monolithic Approach

We begin by discussing the most straightforward approach one could take: a monolithic policy that utilizes all accessible environment states to plan actions for both hands: $\pi_{\text{naive}}(\hat{\xi}_1^b(t), \hat{\xi}_2^b(t), \hat{\xi}_1^h(t), \hat{\xi}_2^h(t) | \xi_1^b(t), \xi_2^b(t), \xi_1^h(t), \xi_2^h(t), o_1(t), o_2(t))$. Training such a policy can be highly inefficient due to the high dimensionality of the observation and action spaces. Impor-

Algorithm 1: AsymDex: Interaction Phase

```
Randomly initialize the two hand bases' poses $\xi_f^b(0)$ and $\xi_d^b(0)$, and initialize object poses
  $o_f(0)$ and $o_d(0)$ based on $\xi_f^b(0)$ and $\xi_d^b(0)$. Initialize policy $\pi_\theta$.
for $iter \in \{1, ..., \max\}$ do
    Initialize replay buffer $\mathcal{B} = \varnothing$ ;
    for $t \in \{1, ..., M\}$ do
        Simulate:
        Collect hand and object states $\xi_f^b(t)$, $\xi_d^b(t)$, $\xi_d^h(t)$, $o_f(t)$, $o_d(t)$;
        Compute relative states $\xi_r^b(t) = \xi_a^b(t) \ominus P_f$, $o_r(t) = o_d(t) \ominus P_f$;
        Policy $\pi_{\mathrm{AsymDex}}(\hat{\xi}_r^b(t), \hat{\xi}_d^h(t) | \xi_r^b(t), \xi_d^h(t), o_r(t))$ outputs relative actions;
        Bimanual controller (Eqn. 1) computes $\hat{\xi}_f^b(t)$ and $\hat{\xi}_d^b(t)$ based on $\hat{\xi}_r^b(t)$;
        if Meet reset condition then
            Reset environment;
        end
        Environment physics steps with $\hat{\xi}_f^b(t), \hat{\xi}_d^b(t), \hat{\xi}_d^h(t)$;
        Evaluate:
        Compute reward $r(t)$
        Collect observations $(\xi_r^b(t), \xi_r^h(t), o_r(t))$, actions $(\hat{\xi}_r^b(t), \hat{\xi}_r^h(t))$, and reward $r(t)$
          into buffer $\mathcal{B}$;
    end
    Update the Policy $\pi_{\mathrm{AsymDex}}$ based on $\mathcal{B}$;
end
Return: Trained policy $\pi_{\mathrm{AsymDex}}$
```

tantly, this monolithic implementation does not exploit the natural asymmetry found in most bimanual tasks. Below, we explain how AsymDex incorporates this insight.

3.3 Incorporating Asymmetry

When humans execute bimanual manipulation tasks, we tend to use our dominant hand to perform precise manipulations, and use a non-dominant hand to facilitate such manipulation [16–18]. For instance, when opening a bottle, we typically use our facilitating hand to move and reorient the bottle such that the bottle cap is closer to and oriented toward the dominant hand, which will then grasp the cap and uncap the bottle. During this cooperation process, our non-dominant hand moves and rotates the object with a firm grasp to facilitate the object manipulation by the dominant hand. Motivated by this, we assign different roles to each robot hand (i.e., a facilitating hand and a dominant hand) during bimanual manipulation.

Further, we make the observation that there tends to be no relative motion between the facilitating hand and the grasped object since the facilitating hand need only hold, move, and reorient the object (i.e., no in-hand reorientation). On the contrary, the hand dominant can interact freely with the object either directly or via another object. This observation suggests that the asymmetric manipulation and coordination strategy is neither dependent on nor influences the hand joints of the facilitating hand. As such, we can considerably reduce the observation and action spaces by accounting for asymmetry. Precisely, this observation allows us to define an asymmetric bimanual policy: $\pi_{\mathrm{asym}}(\hat{\xi}_d^b(t), \hat{\xi}_f^b(t), \hat{\xi}_d^h(t) | \xi_d^b(t), \xi_f^b(t), \xi_d^h(t), o_1(t), o_2(t))$ which use the facilitating hand to only to reposition and reorient the object. This considerable reduction in the dimensions of the observation and action spaces is likely to result in improved sample efficiency.

3.4 Incorporating Relative Observation and Action Spaces

In addition to asymmetry, a key characteristic of bimanual dexterous manipulation is the synchronized motion of the two hands in which each hand moves in response to the other. In this section, we explain how we can further reduce the size of the observation and action spaces by defining relative and object-centric spaces that capture the relationships between the motions of two hands and the object(s) being manipulated. Indeed, the use of relative state spaces has shown to considerably

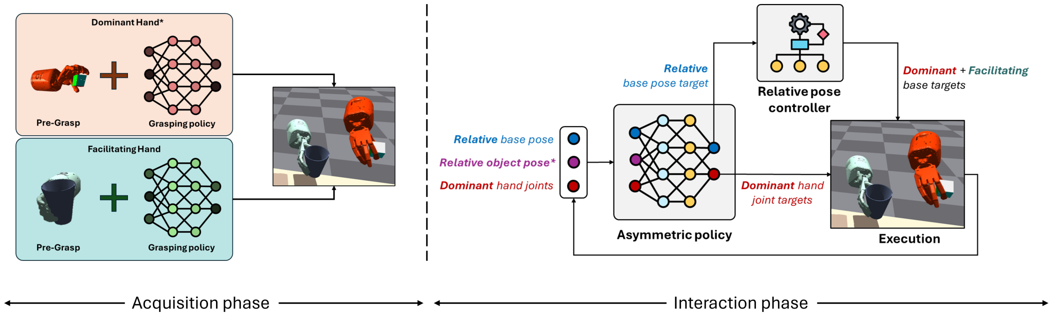


Figure 2: We decompose asymmetric bimanual dexterous manipulation into two phases: An *acquisition* phase and an *interaction* phase. We show that AsymDex can be integrated with learned grasping policies in order to seamlessly execute both phases.

benefit bimanual manipulation with simple end effectors [23, 42–44]. Furthermore, some of these prior work is limiting the relative space in a one-degree-of-freedom (1-DoF) action space [23], while AsymDex allows for complete 6-DoFs relative space.

Let $o_f = o_1$ be the state of the object being held by the facilitating hand, and let $o_d = o_2$ be the state of the object being manipulated by the dominant hand. We attach a coordinate frame to the object being held by the facilitating hand: P_f. Now, we can transform the observations $(\xi_d^b(t), \xi_f^b(t), \xi_d^h(t), o_f(t), o_d(t))$, which where originally defined in the world coordinate frame P_W, into the new coordinate frame P_f. Note that since there is no relative motion between the facilitating hand and the object that it's holding, neither $o_f(t)$ nor $\xi_f^b(t)$ do not change in P_f. As such, both $o_f(t)$ and $\xi_f^b(t)$ can be neglected without losing any information. Now, transforming the remaining observations $(\xi_d^b(t), \xi_d^h(t), o_d(t))$ into the new coordinate P_f yields $(\xi_r^b(t), \xi_d^h(t), o_r(t))$, respectively. Here, $\xi_r^b(t)$ denotes the 6D pose of the dominant hand base and $o_r(t)$ denotes the 6D pose of the object being manipulated by the dominant hand, both now defined relative to the object being held by the facilitating hand. Note that since $\xi_d^h(t)$ denotes the dominant hands' joint states, it is not impacted by the change of coordinates. Similarly, we apply the same modifications to the action spaces of the asymmetry bimanual policy. This allows us to reduce the actions from $(\hat{\xi}_d^b(t), \hat{\xi}_f^b(t), \hat{\xi}_d^h(t))$ to $(\hat{\xi}_r^b(t), \hat{\xi}_d^h(t))$, where $\hat{\xi}_r^b(t)$ is the target relative pose of the dominant hand now defined relative to the object being held by the facilitating hand.

Incorporating the above change of coordinates in addition to leveraging asymmetry, allows us to define AsymDex's policy as $\pi_{\text{AsymDex}}(\hat{\xi}_r^b(t), \hat{\xi}_d^h(t) | \xi_r^b(t), \xi_d^h(t), o_r(t))$. Note that our formulation has significantly reduced the dimensions of both the state and action spaces, compared to the naive policy π_{naive} as defined in Section 3.2. See Appendix. A for RL algorithm and policy architecture.

3.5 Relative Pose Controller

To control the hand bases based on the target relative pose $\hat{\xi}_r^b(t)$ provided by π_{AsymDex}, we designed a bimanual controller that computes both the target dominant hand base pose $\hat{\xi}_d^b(t)$ and the target facilitating hand base pose $\hat{\xi}_f^b(t)$ as follows

$$\begin{aligned}\hat{\xi}_d^b(t) &= \alpha R_{world}^{o_f} \cdot dist(\hat{\xi}_r^b(t), \xi_r^b(t)) + \xi_d^b(t),\\ \hat{\xi}_f^b(t) &= (\alpha - 1) R_{world}^{o_f} \cdot dist(\hat{\xi}_r^b(t), \xi_r^b(t)) + \xi_f^b(t),\end{aligned} \tag{1}$$

where $R_{world}^{o_f}$ denotes the rotational transformation from Frame P_f to the world frame P_W, $dist(\cdot)$ denotes the difference between two 6D poses. and α is a hyperparameter that controls the involvement of each hand. Throughout our experiments, we used fixed $\alpha = 0.5$. The pseudo-code of the training process is included in Alg. 1.

3.6 Acquiring Objects

While our approach as explained thus far deals with the challenge of coordinating two hands to accomplish asymmetric dexterous manipulation tasks, it assumes the task begins with that the object(s) of interest have already been grasped. However, in practice, robots must be able to tackle the challenge of grasping the necessary objects before the interaction between the two hands and the objects can begin. We refer to this phase as the *acquisition* phase. Most recent works on bimanual dexterous manipulation often entirely ignore the acquisition phase and focus purely on the interaction phase [5, 7, 40]. In contrast, we demonstrate that our approach can seamlessly accommodate the acquisition phase by i) leveraging the observation that the acquisition phase doesn't require the coordination of two arms, and ii) employing recent advances in learning to grasp. Specifically, we demonstrate that we can seamlessly integrate AsymDex with PDGM [45], which can efficiently learn multi-fingered grasping policies by leveraging pre-grasp poses (see Fig. 2). Details about the grasping reward design are available in Appendix. B. We begin by executing the grasping policy in isolation and then ”turn on” the asymmetric policy learned by AsymDex after the object has been firmly grasped by the facilitating hand. If the task requires the dominant hand to also grasp a second object, we employ the same method to train a grasping policy for dominant hand to acquire the object, but switch the control of the dominant hand's joints over the asymmetric policy after the object has been grasped.

4 Evaluation

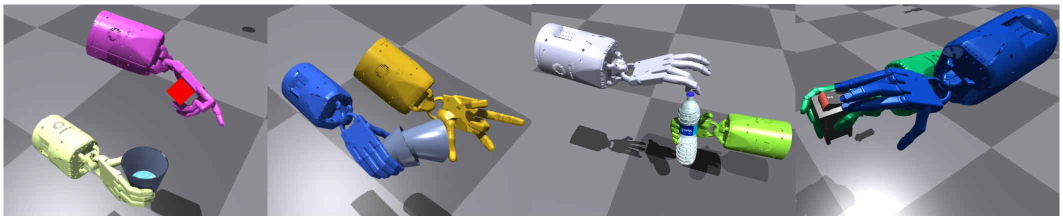
We evaluated AsymDex on four asymmetric bimanual manipulation tasks and compared its performance against strong baselines that challenge our key design choices. We begin by explaining the various aspects of our experimental setup and follow with a discussion of results.

Platform: We conducted all our experiments using two ShadowHands – each a 30-DoF simulated multi-finger hand system (24-DoF hand + 6-DoF floating wrist base) built with Isaac Gym [46].

Tasks: We evaluated AsymDex and the baselines on the following four biannual manipulation tasks which contain both original (*Block in cup*, *Bottle cap*) and adapted tasks (*Stack*, *Switch*) from BiDex-Hand [20] (see Fig. 3 for visualization).

- *Block in cup*: The two hands must coordinate to ensure that one hand places a block inside a cup that is being held by the other without letting either the cup or the block fall to the ground.
- *Stack*: Two cups need to be stacked together. Each hand must hold a cup, and both must coordinate such that the two cups are aligned as one slides into the other.
- *Bottle cap*: One hand must hold and reorient a bottle such that the other hand can grasp and separate the bottle cap from the bottle.
- *Switch*: One hand holds and reorients a switch in a way that allows the other hand to turn it on.

Note that the hands have to coordinate and perform all four tasks without relying on a support surface such as a table and ensure that the object(s) are not dropped. See Appendix. C for details on state space design, sampling procedure, success criteria, and reward design for each task.



(a) Block in cup (b) Stack (c) Bottle cap (d) Switch

Figure 3: We evaluate AsymDex on four bimanual dexterous manipulation tasks.

Metrics: We quantify performance in terms of i) *Task success rate* (see Appendix. C for criteria) and ii) *sample efficiency*. We report both metrics across five random seeds in all experiments.

We evaluated AsymDex with two sets of experiments as described below.

4.1 Learning Bimanual Coordination

In this experiment, we focus on AsymDex's effectiveness during the interaction phase. We initialize the environment such that the hands are at a pre-grasp pose around the objects using appropriate

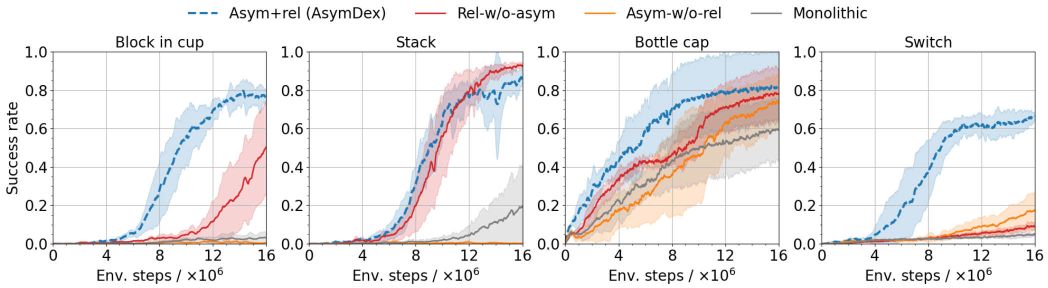


Figure 4: AsymDex consistently outperforms the baselines in terms of sample efficiency and success rate. Solid lines indicate mean trends and shaded areas show ± std. dev., over five random seeds.

Table 1: Success rates (mean ± std. dev.) in Interaction phase

Method **Task**	`Monolithic`	`Asym-w/o-rel`	`Rel-w/o-Asym`	AsymDex (ours)
Block in cup	0.0429 ± 0.0266	0.0164 ± 0.0190	0.3052 ± 0.1577	$\mathbf{0.7701 \pm 0.0559}$
Stack	0.2517 ± 0.2809	0.0044 ± 0.0037	$\mathbf{0.9443 \pm 0.0136}$	0.9233 ± 0.0250
Bottle cap	0.6125 ± 0.1734	0.7893 ± 0.1277	0.8073 ± 0.1389	$\mathbf{0.8301 \pm 0.1797}$
Switch	0.0563 ± 0.0126	0.1626 ± 0.0882	0.1149 ± 0.0176	$\mathbf{0.6700 \pm 0.0359}$

initial grasps (After the first timestep of the environment, the hand need to learn to catch and grasp the object from pre-grasp pose if it is not a facilitating hand). Note that this is a common assumption in recent methods that learn bimanual dexterous manipulation skills [5, 40]. Further, this allows us to isolate and examine AsymDex's ability to learn to coordinate two multi-fingered hands. See Section 4.2 for the second experiment in which we also consider the challenge of acquiring the objects from a tabletop surface before interaction begins.

We compare AsymDex policy against the following baselines:

- `Monolithic`: This policy doesn't make assumptions about the structure of bimanual manipulation (see Sec. 3.2). As such, this baseline allows us to examine the necessity and effectiveness of leveraging both the asymmetry in hand roles and the relative action and observation spaces.
- `Asym-w/o-rel`: This policy leverages asymmetry in hand roles, but learns over absolute observation and action places (see Sec. 3.3). As such, this baseline allows us to examine the necessity and effectiveness of relative action and observation spaces.
- `Rel-w/o-asym`: This policy leverages the relative observation and action places, but ignores asymmetry. As such, it allows us to examine the necessity and effectiveness of asymmetry.

We report the learning curves in Fig. 4 and success rates in Table 1. As can be readily observed, AsymDex is the only method that consistently performs well across all tasks, either performing comparably or outperforming all the baselines. While `Rel-w/o-asym` performs comparably to AsymDex on *Stack*, it is not able to match AsymDex's performance on the other tasks. Both AsymDex and `Rel-w/o-asym` perform substantially better than the other two baselines across all tasks except *Switch*, where AsymDex performs much better than all three baselines. In fact, `Asym-w/o-rel` is able to learn an effective policy in only one task (*Bottle cap*), while `Monolithic` struggles on all four tasks. Curiously, `Monolithic` outperforms `asym-w/o-rel` on the *Stack* task. Qualitative analysis of rollouts reveals that, unlike `Asym-w/o-rel`, the `Monolithic` policy learns to use the facilitating hand's fingers to orient the cup towards the bottom of the other cup.

Taken together, the above observations reveal a few insights. First, when used in isolation, neither asymmetry nor relative motion are sufficient across all tasks. In particular, while each might provide sufficient structure to accomplish some tasks, they prove to be less effective on other tasks. Second, the use of relative motion offers a larger boost in performance compared to asymmetry, likely due to the fact that relative spaces avoid unnecessary exploration (e.g., when the two hands move in parallel) while allowing the facilitating hand to exhibit more complex behaviors. Third, ignoring both asymmetry and relative motion hardly leads to success.

4.2 Learning to Grasp and Coordinate

In this experiment, we evaluate AsymDex's ability to incorporate the object acquisition phase in addition to the interaction phase. Specifically, we initialize the environment for each task such that the objects of interest are placed on a tabletop surface. As such, each method needs to learn both to grasp the necessary objects and to coordinate the two hands to complete the tasks.

For AsymDex, we follow the same strategy as introduced in Section 3.6. We compare AsymDex's performance against the following baselines:

- `1-stage-monolithic`: This baseline uses a single policy to learn both the grasping and interaction phases for both hands, and thus allows us to investigate the benefits of AsymDex's two-phase decomposition.
- `2-stage-monolithic`: This policy benefits from the same phase decomposition as AsymDex policy, but leverages neither asymmetry nor relative motion. As such, this baseline allows us to examine if this task can be solved merely with two-phase decomposition.

To ensure a fair comparison, we provide pre-grasp pose annotations to both baselines. Further, we ensure that the total number of env. interactions (the number of one stage or the sum of two stages) is the same across AsymDex and the baselines. See Appendix. B for details of the grasping learning.

We report the overall roll-out success rate of all methods for two tasks across five random seeds in Table. 2. We find that AsymDex significantly outperforms the other two baselines in both tasks, suggesting that combining phase decomposition with AsymDex's other two design choices (asymmetry and relative state) results in policies that can effectively handle both the acquisition and the interaction phases of bimanual dexterous manipulation. The fact that `2-stage-monolithic` baseline outperforms the `1-stage-monolithic` baseline demonstrates the inherent benefits of phase decomposition. Our qualitative analysis of *Block in cup* task revealed that `1-stage-monolithic` policy learns to tip the cup over and push the block towards the cup. In contrast, both the two-stage policies learn the expected behavior. This suggests that the phase decomposition nudges the grasping and interaction policies to learn reasonable behaviors that complement each other.

Table 2: Success rates (mean ± std. dev.) after combining acquisition and interaction phases

Method **Task**	`1-stage-monolithic`	`2-stage-monolithic`	`2-stage-AsymDex`
Block in cup	0.0321 ± 0.0251	0.1505 ± 0.1059	$\mathbf{0.7938 \pm 0.0897}$
Bottle cap	0.1680 ± 0.2695	0.6407 ± 0.1141	$\mathbf{0.8726 \pm 0.0600}$

5 Conclusion

Our framework (AsymDex) is capable of learning complex asymmetric bimanual dexterous manipulation tasks via reinforcement without relying on demonstrations. We introduced and validated the need for AsymDex's two crucial ingredients: assigning asymmetric roles to the two hands, and using relative observation and action spaces. Our evaluation results reveal that the combination of these choices consistently leads to better sample efficiency and success rates across different tasks.

6 Limitations and Future Works

Our work has revealed a number of limitations and avenues for future research. First, AsymDex in its current form cannot handle certain bimanual tasks that require complex multi-finger manipulation from both hands (e.g., reorienting a heavy object, dynamic handover). Second, AsymDex does not consider the kinodynamic constraints that might result from manipulator arms. Third, AsymDex has not yet been evaluated on hardware. Fourth, behaviors produced by AsymDex are not always natural or human-like due to lack of necessary incentives.

References

[1] C. Smith, Y. Karayiannidis, L. Nalpantidis, X. Gratal, P. Qi, D. V. Dimarogonas, and D. Kragic. Dual arm manipulation—a survey. *Robotics and Autonomous systems*, 60(10):1340–1353, 2012.

[2] J. Grannen, Y. Wu, B. Vu, and D. Sadigh. Stabilize to act: Learning to coordinate for bimanual manipulation. In *Conference on Robot Learning*, pages 563–576. PMLR, 2023.

[3] Y. Avigal, L. Berscheid, T. Asfour, T. Kröger, and K. Goldberg. Speedfolding: Learning efficient bimanual folding of garments. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2022.

[4] C. Chi, Z. Xu, C. Pan, E. Cousineau, B. Burchfiel, S. Feng, R. Tedrake, and S. Song. Universal manipulation interface: In-the-wild robot teaching without in-the-wild robots. *arXiv preprint arXiv:2402.10329*, 2024.

[5] B. Huang, Y. Chen, T. Wang, Y. Qin, Y. Yang, N. Atanasov, and X. Wang. Dynamic handover: Throw and catch with bimanual hands. *arXiv preprint arXiv:2309.05655*, 2023.

[6] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation. *arXiv preprint arXiv:2403.07788*, 2024.

[7] K. Zakka, P. Wu, L. Smith, N. Gileadi, T. Howell, X. B. Peng, S. Singh, Y. Tassa, P. Florence, A. Zeng, et al. Robopianist: Dexterous piano playing with deep reinforcement learning. *arXiv preprint arXiv:2304.04150*, 2023.

[8] A. Rajeswaran, V. Kumar, A. Gupta, G. Vezzani, J. Schulman, E. Todorov, and S. Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations. In *Proceedings of Robotics: Science and Systems (RSS)*, 2018.

[9] OpenAI, M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning Dexterous In-Hand Manipulation. *International Journal of Robotics Research (IJRR)*, 2020.

[10] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik. In-hand object rotation via rapid motor adaptation. *arXiv preprint arXiv:2210.04887*, 2022.

[11] Y. Han, M. Xie, Y. Zhao, and H. Ravichandar. On the utility of koopman operator theory in learning dexterous manipulation skills. In *Conference on Robot Learning*, pages 106–126. PMLR, 2023.

[12] Y. Han, Z. Chen, K. A. Williams, and H. Ravichandar. Learning prehensile dexterity by imitating and emulating state-only observations. *IEEE Robotics and Automation Letters*, 9(10): 8266–8273, 2024.

[13] H. Chen, A. ABUDUWEILI, A. Agrawal, Y. Han, H. Ravichandar, C. Liu, and J. Ichnowski. Korol: Learning visualizable object feature with koopman operator rollout for manipulation. In *8th Annual Conference on Robot Learning*.

[14] K. Shaw, S. Bahl, A. Sivakumar, A. Kannan, and D. Pathak. Learning dexterity from human hand motion in internet videos. *The International Journal of Robotics Research*, page 02783649241227559, 2024.

[15] Y. Guiard. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of motor behavior*, 19(4):486–517, 1987.

[16] M. Kimmerle, C. L. Ferre, K. A. Kotwica, and G. F. Michel. Development of role-differentiated bimanual manipulation during the infant's first year. *Developmental Psychobiology: The Journal of the International Society for Developmental Psychobiology*, 52(2):168–180, 2010.

[17] R. L. Sainburg. Evidence for a dynamic-dominance hypothesis of handedness. *Experimental brain research*, 142:241–258, 2002.

[18] B. E. Studenka and H. N. Zelaznik. The influence of dominant versus non-dominant hand on event and emergent motor timing. *Human Movement Science*, 27(1):29–52, 2008.

[19] L. Cashmore, N. Uomini, and A. Chapelain. The evolution of handedness in humans and great apes: a review and current issues. *Journal of anthropological sciences*, 86(2008):7–35, 2008.

[20] Y. Chen, T. Wu, S. Wang, X. Feng, J. Jiang, Z. Lu, S. McAleer, H. Dong, S.-C. Zhu, and Y. Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning. *Advances in Neural Information Processing Systems*, 35:5150–5163, 2022.

[21] G. Franzese, L. de Souza Rosa, T. Verburg, L. Peternel, and J. Kober. Interactive imitation learning of bimanual movement primitives. *IEEE/ASME Transactions on Mechatronics*, pages 1–13, 2023.

[22] M. Seo, S. Han, K. Sim, S. H. Bang, C. Gonzalez, L. Sentis, and Y. Zhu. Deep imitation learning for humanoid loco-manipulation through human teleoperation. In *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*, pages 1–8. IEEE, 2023.

[23] A. Bahety, P. Mandikal, B. Abbatematteo, and R. Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. *arXiv preprint arXiv:2405.03666*, 2024.

[24] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[25] Y. Lin, A. Church, M. Yang, H. Li, J. Lloyd, D. Zhang, and N. F. Lepora. Bi-touch: Bimanual tactile manipulation with sim-to-real deep reinforcement learning. *IEEE Robotics and Automation Letters*, 2023.

[26] S. Kataoka, S. K. S. Ghasemipour, D. Freeman, and I. Mordatch. Bi-manual manipulation and attachment via sim-to-real reinforcement learning. *arXiv preprint arXiv:2203.08277*, 2022.

[27] R. Chitnis, S. Tulsiani, S. Gupta, and A. Gupta. Efficient bimanual manipulation using learned task schemas. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1149–1155. IEEE, 2020.

[28] Y. Li, C. Pan, H. Xu, X. Wang, and Y. Wu. Efficient bimanual handover and rearrangement via symmetry-aware actor-critic learning. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3867–3874. IEEE, 2023.

[29] H. Jo, W. Choi, G. Lee, W. Park, and J. Kim. Analysis of visuo motor control between dominant hand and non-dominant hand for effective human-robot collaboration. *Sensors*, 20(21):6368, 2020.

[30] R. Holladay, T. Lozano-Pérez, and A. Rodriguez. Robust planning for multi-stage forceful manipulation. *The International Journal of Robotics Research*, 43(3):330–353, 2024.

[31] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh. Learning bimanual scooping policies for food acquisition. *arXiv preprint arXiv:2211.14652*, 2022.

[32] J. Liu, Y. Chen, Z. Dong, S. Wang, S. Calinon, M. Li, and F. Chen. Robot cooking with stir-fry: Bimanual non-prehensile manipulation of semi-fluid objects. *IEEE Robotics and Automation Letters*, 7(2):5159–5166, 2022.

[33] Y. Cui, Z. Xu, L. Zhong, P. Xu, Y. Shen, and Q. Tang. A task-adaptive deep reinforcement learning framework for dual-arm robot manipulation. *IEEE Transactions on Automation Science and Engineering*, 2024.

[34] A. Nagabandi, K. Konolige, S. Levine, and V. Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020.

[35] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang. Dexmv: Imitation learning for dexterous manipulation from human videos. In *European Conference on Computer Vision*, pages 570–587. Springer, 2022.

[36] G. Khandate, S. Shang, E. T. Chang, T. L. Saidi, J. Adams, and M. Ciocarlie. Sampling-based Exploration for Reinforcement Learning of Dexterous Manipulation. In *Proceedings of Robotics: Science and Systems*, Daegu, Republic of Korea, July 2023. doi:10.15607/RSS.2023.XIX.020.

[37] M. Xie, A. Handa, S. Tyree, D. Fox, H. Ravichandar, N. D. Ratliff, and K. Van Wyk. Neural geometric fabrics: Efficiently learning high-dimensional policies from demonstration. In *Conference on Robot Learning*, pages 1355–1367. PMLR, 2023.

[38] A. Handa, K. Van Wyk, W. Yang, J. Liang, Y.-W. Chao, Q. Wan, S. Birchfield, N. Ratliff, and D. Fox. Dexpilot: Vision-based teleoperation of dexterous robotic hand-arm system. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9164–9170. IEEE, 2020.

[39] S. P. Arunachalam, I. Güzey, S. Chintala, and L. Pinto. Holo-dex: Teaching dexterity with immersive mixed reality. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5962–5969. IEEE, 2023.

[40] T. Lin, Z.-H. Yin, H. Qi, P. Abbeel, and J. Malik. Twisting lids off with two hands. *arXiv preprint arXiv:2403.02338*, 2024.

[41] T. Chen, E. Cousineau, N. Kuppuswamy, and P. Agrawal. Vegetable peeling: A case study in constrained dexterous manipulation. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition@ CoRL2023*, 2023.

[42] R. Laha, J. Vorndamme, L. F. Figueredo, Z. Qu, A. Swikir, C. Jähne, and S. Haddadin. Coordinated motion generation and object placement: A reactive planning and landing approach. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9401–9407. IEEE, 2021.

[43] P. Chiacchio, S. Chiaverini, and B. Siciliano. Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. 1996.

[44] S. Tarbouriech, B. Navarro, P. Fraisse, A. Crosnier, A. Cherubini, and D. Sallé. Dual-arm relative tasks performance using sparse kinematic control. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6003–6009. IEEE, 2018.

[45] S. Dasari, A. Gupta, and V. Kumar. Learning dexterous manipulation from exemplar object trajectories and pre-grasps. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3889–3896. IEEE, 2023.

[46] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox. Gpu-accelerated robotic simulation for distributed reinforcement learning, 2018.

[47] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms, 2017.

Appendices

A RL Training

We use Proximal Policy Optimization (PPO) [47] algorithm to train all policies $\pi_{\text{naive}}, \pi_{\text{asym}}$, and π_{AsymDex} with their corresponding value functions. Both policies and value functions are parameterized via a three-layer MLP network. The size of hidden layers for each is i) policy: (256, 256, 128), ii) value function: (512, 512, 512). The activation functions are all set as Exponential Linear Unit (ELU). We use the same PPO hyperparameters for all the baselines and AsymDex ($learning\ rate : 3 \times 10^{-4}, \gamma : 0.98, \lambda : 0.95$). We train the polices on a computer with a single Nividia RTX 4090 GPU.

B Grasping Learning

Two-stage policy Both AsymDex (our approach) and `2-stage-monolithic` policy (one of the baselines in Sec. 4.2) are two-stage policies. Therefore, they can first learn a grasping policy for the facilitating hand (or two grasping policies for facilitating hand and dominant hand respectively). Such policy $\pi_{\text{grasp}}(\hat{\xi}_f^h(t)|\xi_f^h(t), \xi_f^b(t) \ominus X_{o_f})$ takes in the hand joint states and the relative pose between hand and the object, and outputs the target hand joint positions to grasp the object firmly. We first provide pre-grasp annotations [45], which allows the hands to initialize at the position close to the objects with proper joint positions. Then we script the $6D$ lifting hand base motions and design the the following rewards, which is the same across all objects.

$$Reward = R_{rel\ pos} + R_{rel\ rot}$$

The relative position reward $R_{rel\ pos} = (\alpha - ||x_{obj} - x_{initial}||) * \beta$, where x_{obj} is the current relative position between the object and the hand, and $x_{initial}$ is the initial relative position between the object and the hand. The $\alpha, \beta \in \mathcal{R}_+$ are hyper-parameters. The relative rotation reward $R_{rel\ pos} =< u_{obj}, u_{hand} >$, where u_{obj} is the object direction vector, u_{hand} is the hand direction vector, and $< \cdot, \cdot >$ denotes the inner product of two vectors. We define the object direction vector and hand direction vector to be the same at the beginning of the grasping phase. Both rewards encourage the hand to keep a constant relative pose, i.e., grasping the object, during the script motion.

One-stage policy Another baseline in Sec. 4.2, i.e., the `1-stage-monolithic` policy, does not incorporate the task decomposition. Therefore, it only uses the task-specific interaction rewards (see Appendix. C) to learn how to complete the entire bimanual task. For a fair comparison, both hands also start at the pre-grasp poses.

C Task Design

In this section, we show the details for each task.

State Space Design For each task, the hand joint states $\xi_f^h(t)$, $\xi_d^h(t)$ include the 24-DoF hand joint positions and the 24-DoF hand joint velocities. We use quaternions to represent rotation part of object and hand base poses. And for all policies, we also include the previous actions in the policy input. For the *block in cup* task, $o_f(t)$ and $o_d(t)$ represent the poses of the cup and the block respectively. For the *stack* task, $o_f(t)$ and $o_d(t)$ represent the poses of two cups. For the *Bottle cap* task, $o_f(t)$ and $o_d(t)$ represent the poses of the bottle and the cap respectively. For the *Switch* task, $o_f(t)$ and $o_d(t)$ represent the poses of the switch body and the button respectively. The dimensions of the observation and action spaces of each policy are shown in Table. 3. It is obvious that AsymDex policy significantly reduces the state dimensions.

Sampling Procedure

- *Place block in cup*: The initial position of dominant hand base is randomized: $x_d \in \mathcal{X} \sim \mathcal{U}(0.3, 0.7)$, $y_d \in \mathcal{Y} \sim \mathcal{U}(-0.2, 0.0)$, $z_d \in \mathcal{Z} \sim \mathcal{U}(0.7, 1.1)$. For the rotation of the dominant hand, we randomly rotate it around the axis along the arm at a random angle,

Table 3: Dimension of Observation and action spaces. For all tasks, the dimensions are identical.

	`Monolithic`	`Asym-w/o-rel`	`Rel-w/o-Asym`	`AsymDex` (ours)
Observation	176	108	163	88
Action	52	32	46	26

$\alpha \in \mathcal{A} \sim \mathcal{U}(-1.57, 1.57)$, in radians. The block is initialized in the dominant hand. Thus its position and rotation is calculated based on the initial position and rotation of dominant hand base. The initial position of the facilitating hand base is at $[0.55, 0.6, 0.8]$.

- *Stack cups*: The initial position of dominant hand base is randomized: $x_d \in \mathcal{X} \sim \mathcal{U}(0.3, 0.7)$, $y_d \in \mathcal{Y} \sim \mathcal{U}(-0.2, 0.0)$, $z_d \in \mathcal{Z} \sim \mathcal{U}(0.7, 1.1)$. For the rotation of the dominant hand, we randomly rotate it around the axis along the arm at a random angle, $\alpha \in \mathcal{A} \sim \mathcal{U}(-1.57, 1.57)$, in radians. The cup is initialized in the dominant hand. Thus its position and rotation is calculated based on the initial position and rotation of dominant hand. The initial position of the facilitating hand base is at $[0.55, 0.6, 0.8]$.
- *Open bottle cap*: The initial position of dominant hand base is randomized: $x_d \in \mathcal{X} \sim \mathcal{U}(0.58, 0.62)$, $y_d \in \mathcal{Y} \sim \mathcal{U}(-0.21, -0.19)$, $z_d \in \mathcal{Z} \sim \mathcal{U}(0.58, 0.62)$. For the rotation of the dominant hand, we randomly rotate it around the axis along the arm at a random angle, $\alpha \in \mathcal{A} \sim \mathcal{U}(-1.0, 1.0)$, in radians. The initial position of the facilitating hand base is randomized: $x_f \in \mathcal{X} \sim \mathcal{U}(0.53, 0.57), y_f \in \mathcal{Y} \sim \mathcal{U}(0.59, 0.61), z_f \in \mathcal{Z} \sim \mathcal{U}(0.43, 0.45)$. For the rotation of the facilitating hand, we randomly rotate it around the axis along the arm at a random angle, $\beta \in \mathcal{B} \sim \mathcal{U}(-0.5, 0.5)$, in radians. The bottle is initialized in the facilitating hand. Thus its position and rotation is calculated based on the initial position and rotation of facilitating hand base.
- *Turn on switch*: The initial position of dominant hand base is randomized: $x_d \in \mathcal{X} \sim \mathcal{U}(0.2, 0.6)$, $y_d \in \mathcal{Y} \sim \mathcal{U}(-0.25, -0.05)$, $z_d \in \mathcal{Z} \sim \mathcal{U}(0.5, 0.9)$. For the rotation of the dominant hand, we randomly rotate it around the axis along the arm at a random angle, $\alpha \in \mathcal{A} \sim \mathcal{U}(-1.0, 1.0)$, in radians. The initial position of the facilitating hand base is randomized: $x_f \in \mathcal{X} \sim \mathcal{U}(0.2, 0.6)$, $y_f \in \mathcal{Y} \sim \mathcal{U}(0.05, 0.25)$, $z_f \in \mathcal{Z} \sim \mathcal{U}(0.41, 0.81)$. For the rotation of the facilitating hand, we randomly rotate it around the axis along the arm at a random angle, $\beta \in \mathcal{B} \sim \mathcal{U}(-1.0, 1.0)$, in radians. The switch is initialized in the facilitating hand. Thus its position and rotation is calculated based on the initial position and rotation of facilitating hand base.

Success Criteria

- *Place block in cup* The task is considered successful if the distance of the block center and the cup center is smaller than 0.035 meters. This distance makes sure the task is only considered successful when the block is inside the cup. If the block falls on the ground or has not entered the cup within a certain time step, the task is considered failed.
- *Stack cups* The task is considered successful if the distance between the cup centers is smaller than 0.02 meters. If either cup falls on the ground or has not been stacked within a certain time step, the task is considered failed.
- *Open bottle cap* The task is considered successful if the cap is taken off from its original position 0.05 meters away within a time duration, and is considered failed otherwise.
- *Turn on switch* The button and the switch body are connected by a revolute joint ranging from 0 to 0.5585 rads. The task is considered successful if the button is pressed and rotated 0.3585 rads within a time duration, and is considered failed otherwise.

Reward Design The reward design is similar across all tasks:

$$Reward = R_{hand\ distance} + R_{progress} + R_{action\ penalty} + R_{success\ bonus}$$

For each task, $R_{action\ penalty} = -||a(t)||^2$, and the $R_{success\ bonus}$ is the task success reward. $R_{hand\ distance}$ and $R_{progress}$ are slightly different for each tasks.

- *Place block in cup*: $R_{hand\ distance} = e^{-||x_{palm} - x_{cup\ mouth}||}$, where x_{palm} is the dominant hand palm position, and $x_{cup\ mouth}$ is the position of the cup mouth. $R_{progress} = -||x_{cup} - x_{block}||$, where x_{cup} is the position of the cup, and x_{block} is the position of the block.

- *Stack cups*: $R_{hand\ distance} = e^{-||x_{palm} - x_{cup\ mouth}||}$, where x_{palm} is the dominant hand palm position, and $x_{cup\ mouth}$ is the position of the cup mouth, which is grasped by the facilitating hand. $R_{progress} = -||x_{cupd} - x_{cupf}||$, where x_{cupf} is the position of the cup grasped by the facilitating hand, and x_{cupd} is the position of the cup grasped by the dominant hand.
- *Open bottle cap*: $R_{hand\ distance} = (1 - (||x_{index} - x_{cap}|| + ||x_{thumb} - x_{cap}||))^3$, where x_{index} and x_{thumb} are the tip position of index finger and thumb respectively, and x_{cap} is the position of the bottle cap. $R_{progress} = ||x_{cap} - x_{bottle\ top}||$, where x_{cap} is the position of the cap, and $x_{bottle\ top}$ is the position of the top of the bottle.
- *Turn on switch*: $R_{hand\ distance} = (1 - (||x_{index} - x_{button}|| + ||x_{thumb} - x_{button}||))^3$, where x_{index} and x_{thumb} are the tip position of index finger and thumb respectively, and x_{button} is the position of the button. $R_{progress} = 2 * \theta_{button}$, where θ_{button} is the rotated angle of the joint that connects the button and the switch body.